

GMSEC

GSFC Mission Services Evolution Center



At A Glance

The API provides a common interface to the GMSEC software information bus.

Features

- Standardized messages formats
- Standardized communications interface
- Multiple language, middleware, and operating system support

Benefits

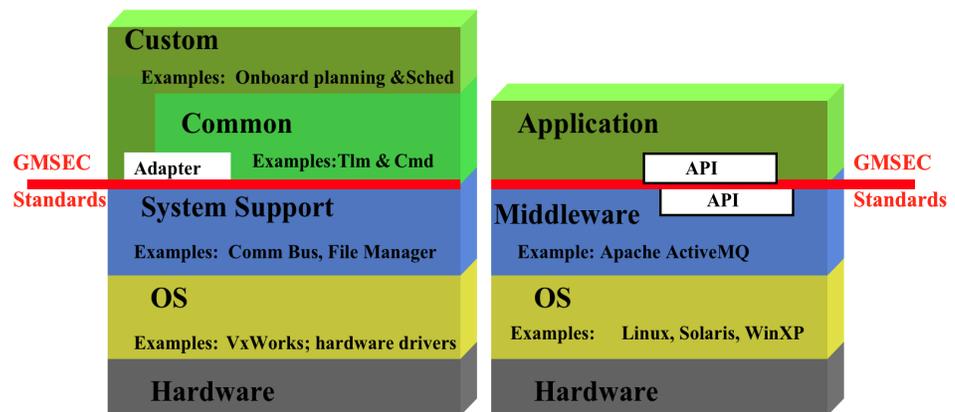
- Isolates both complexity of applications and communications management
- Simplifies applications development
- Simplifies integration
- Facilitates plug-and-play and portability of applications
- Eliminates middleware dependencies



GMSEC Application Programming Interface (API)

Overview

The Goddard Mission Services Evolution Center (GMSEC) architecture is a comprehensive flight and ground data system framework that supports the full mission lifecycle. The GMSEC API provides a mechanism by which the diverse set of flight and ground software components can easily exchange data or messages. Software components using the GMSEC API are connected to the middleware which in turn is responsible for message routing. This frees the components from having to know where other components exist and what data the other components need. The key role of the GMSEC architecture is to define and manage these GMSEC messages through standards to ensure seamless component upgrades, exchanges, or reconfigurations with minimal integration.



GMSEC Layered Architecture for Flight and Ground Systems

GMSEC Message Exchange

The application-to-middleware interface uses subject-based addressing. Each application provides its output products to the middleware, and requests the data it needs from the middleware through a standard interface. The application has no knowledge that any other specific application exists, or on what platform it might reside. This communications model is often described as publish/subscribe; the application publishes its products to the middleware for all other applications to receive, and subscribes to the middleware for the data it requires, whatever the source. Applications subscribe for messages by providing the requested message subject to the middleware. Applications publish messages by providing a GMSEC standard message and the message's associated subject to the middleware. The middleware takes the responsibility of routing a copy of the message with that subject to all requesting applications. By careful subject name delineation, messages can be sent to a single application, group(s), or all components. Subject names with both fixed and variable elements within their constructs allow the flexibility to meet a variety of message distribution requirements.

**NASA Goddard Mission Services Evolution Center, Code 581
Greenbelt, Maryland 20771**

<http://gmsec.gsfc.nasa.gov>

email: GMSEC@nasa.gov

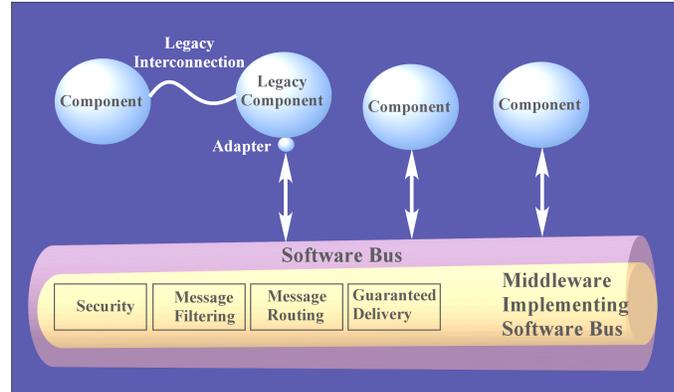
Middleware

The middleware provides real-time asynchronous interprocess communications to move messages between applications. The middleware assumes the responsibility for the details of the communications transport network, such as publisher and subscriber location, message routing, network protocols, data marshalling and message queuing to deliver each message. The details of the message communication are isolated behind the application middleware interface in order to provide the maximum flexibility to incorporate different vendor products and new technologies within the middleware. The middleware component is intentionally independent of the traditional space mission functional capabilities. This insulation between the space mission applications and the middleware allows commercial middleware solutions (designed for a larger and more general marketplace) to be used in the GMSEC architecture.

Systems Supported

The GMSEC API was initially developed for a range of popular computer platforms and software development languages, and for use with a limited number of underlying middleware vendor products. Support for additional platforms, languages, and middleware will be driven by the requirements of future missions, components, and users. The table below shows the toolsets currently supported. Development teams are provided with the APIs, installation packages, supporting documentation, and test routines. Further technical support is provided by the GMSEC team and by the product vendors.

Category	Products Currently Supported
Programming Languages	C, Java C++, Perl
Operating Systems	Microsoft Windows 2003/XP/7 Red Hat Linux Solaris
Compilers	Linux: GNU C++ Compiler (GCC) Microsoft: Visual Studio 2008, 2010 Solaris: GCC, CC
Middleware	GMSEC Message Bus, TIBCO SmartSockets, IBM WebSphere MQ, Apache ActiveMQ, Oracle WebLogic (C/C++, Perl)



Components Interface to the GMSEC Software Bus

API Function Details

The GMSEC API specification has been instantiated for a variety of programming languages. Each instance of the API for a programming language provides a set of common message communication functions with both synchronous and asynchronous delivery mechanisms:

Connection Functions

Connect, Disconnect: Initiate and terminate a connection between the application and the middleware

Build a Message Functions

CreateMessage, DestroyMessage: Allocate and release buffer where message content is built

Publish/Subscribe Functions

Publish: Send a message to all subscribers on the previously established connection.

Subscribe, Unsubscribe: Registers and unregisters a subject name to receive messages over the connection

Request/Reply Functions

Request, Reply: Sends a request message on the connection. The receiver will return a message with the Reply function.

Message Retrieval Functions

GetNextMsg, DispatchMsg, StartAutoDispatch, StopAutoDispatch: Used by a subscriber to pull received messages from the incoming queue.

